地球系统数值模拟装置项目（地球系统模式数值模拟系统）集成模块分系统培训 CAS-ESM耦合

培训人：何卷雄

2022年5月31日

内容

01 背景

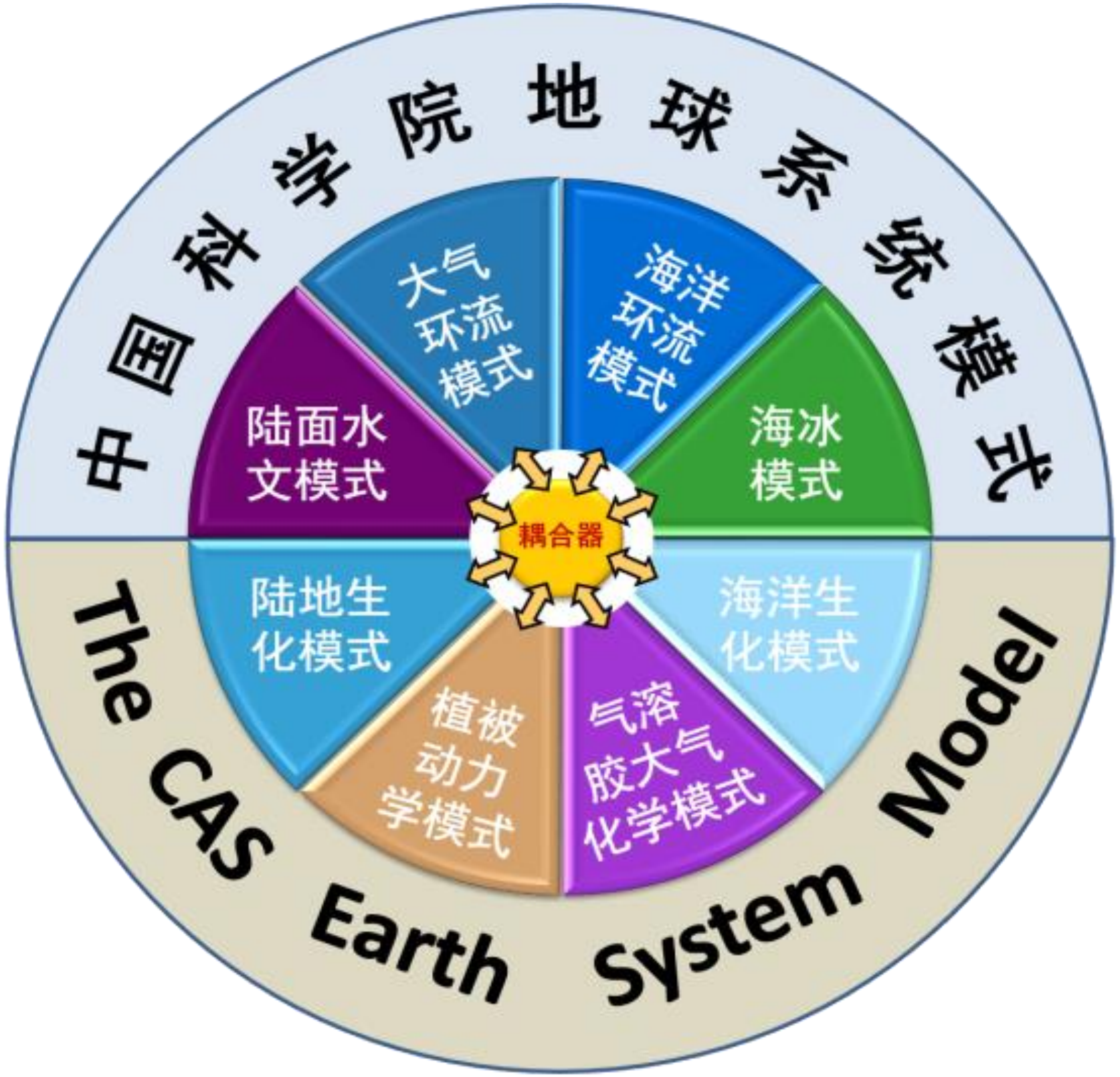
02 耦合器结构

03 耦合器机制

04 耦合器工作流程

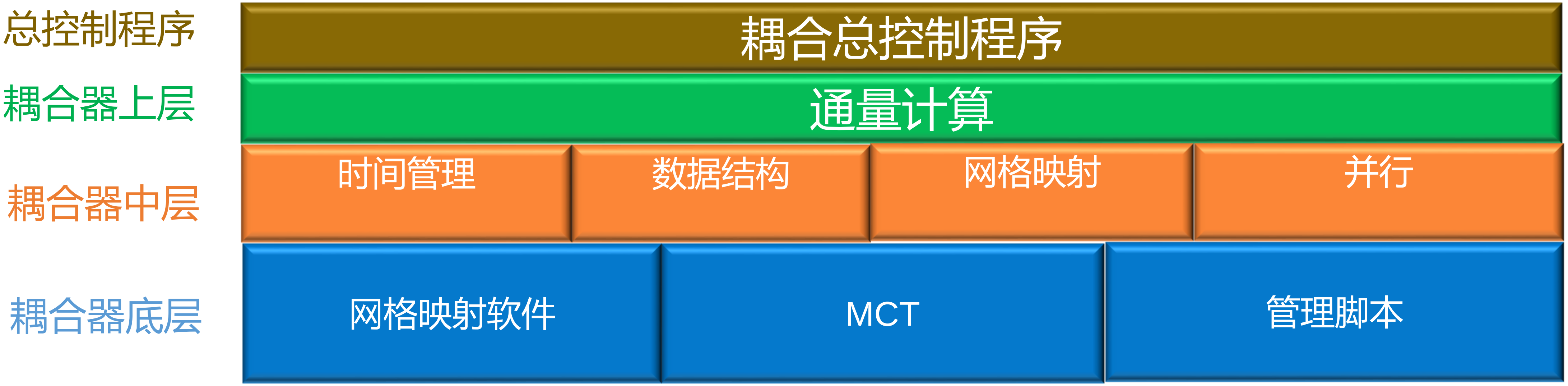
05 分量模式数据交换

背景



- 基于CPL7
- 软件工程需要
  - 历史代码继承
  - 团队协作
  - 任务自然属性
- 超级计算机架构需要
  - 同构平台
  - 节点内共享内存
  - MPI，SPMD

耦合器结构



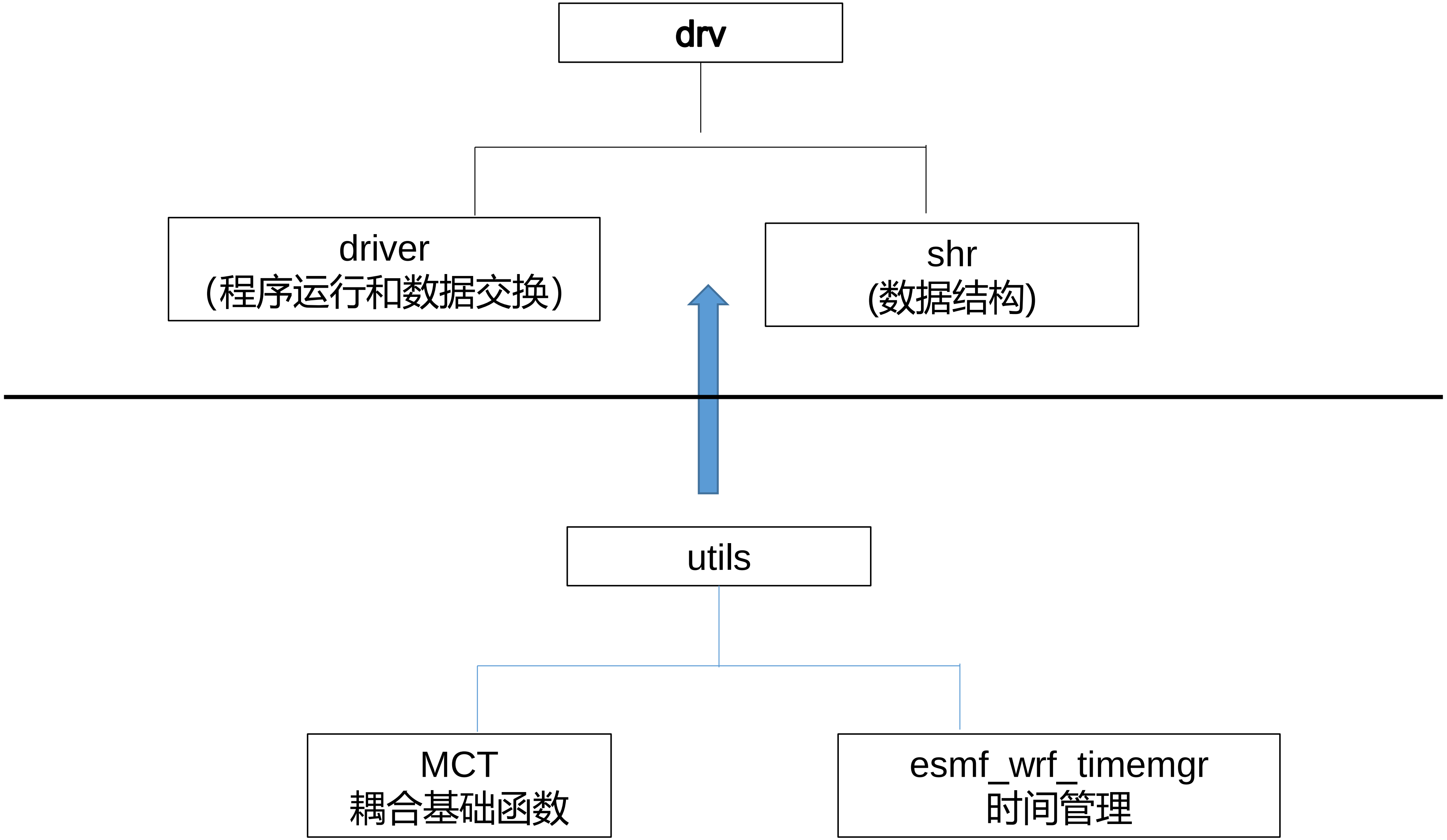
总控制程序 耦合总控制程序

耦合器上层 通量计算

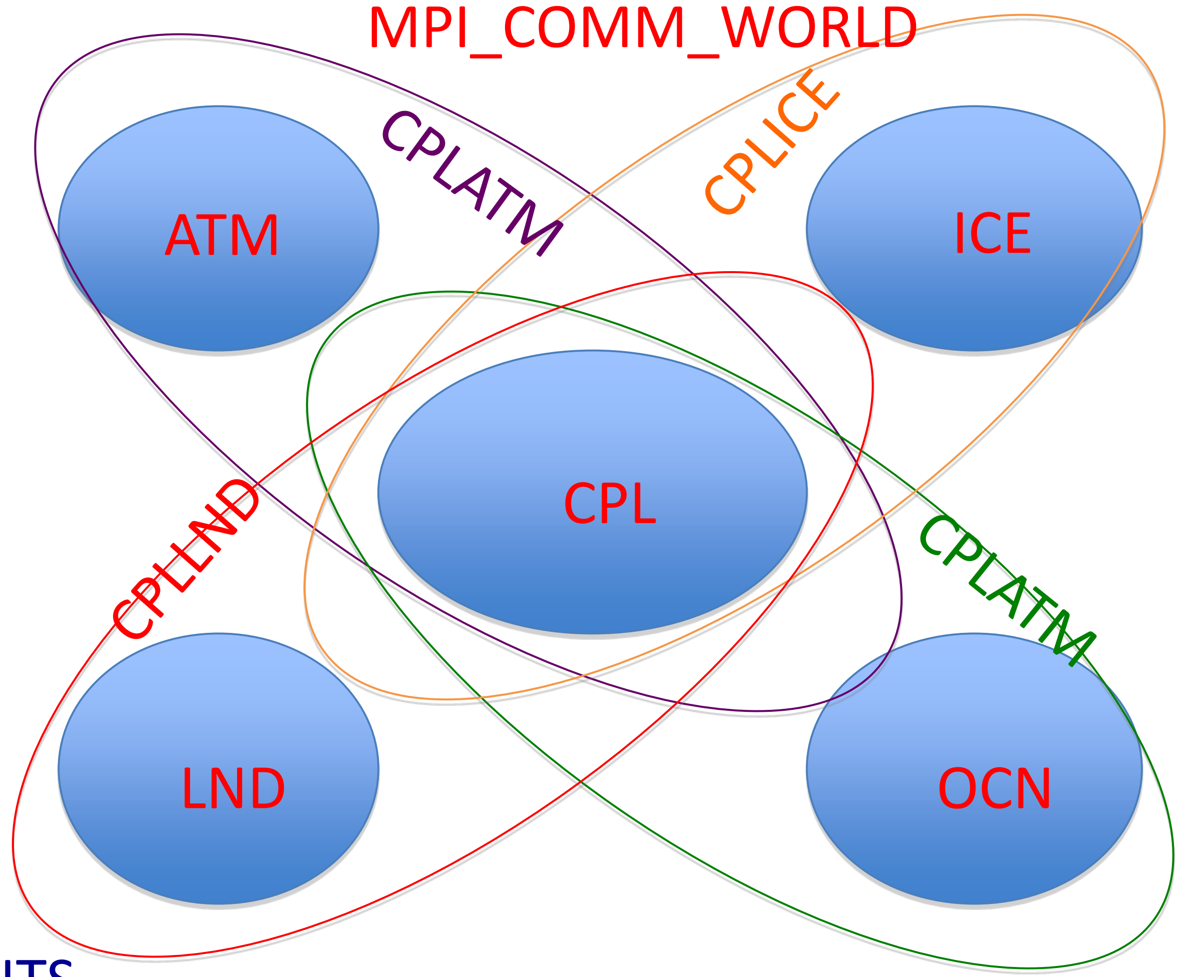
耦合器中层 时间管理 数据结构 网格映射 并行

耦合器底层 网格映射软件 MCT 管理脚本

CPL7代码结构


drv
driver
(程序运行和数据交换)
shr
(数据结构)
utils
MCT
耦合基础函数
esmf_wrf_timemgr
时间管理

并行机制

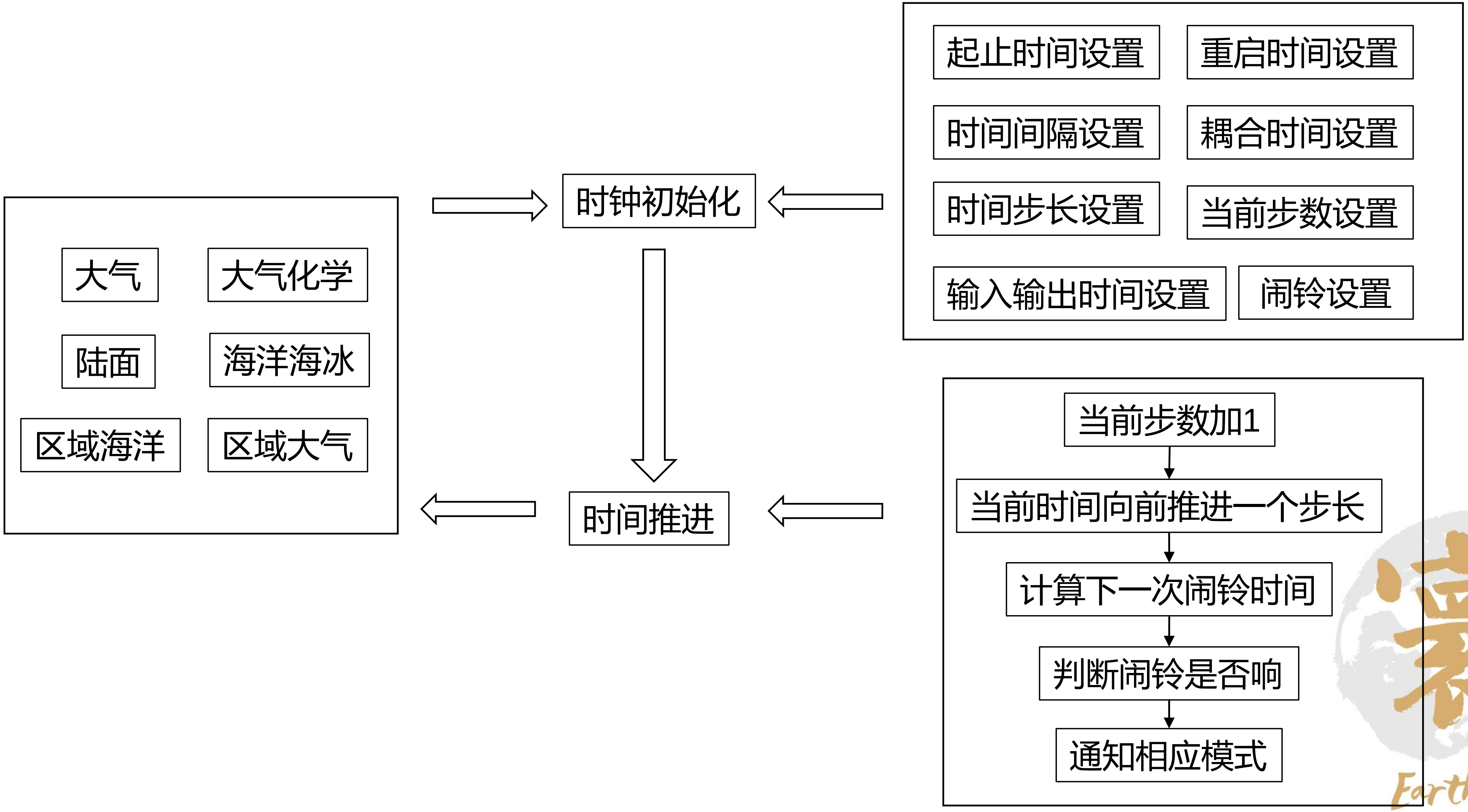


5 COMPONENTS
10 COMMUNICATION GROUPS

并行机制

```
! Initialize
call mpi_init(ierr)
call mpi_comm_rank(MPI_COMM_WORLD, mype  , ierr)
call mpi_comm_size(MPI_COMM_WORLD, numpes, ierr)

! Create MPI communicator groups
call seq_comm_setcomm(ATMID,…) ! Atmoshere
call seq_comm_setcomm(LNDID, …) ! land
call seq_comm_setcomm(ICEID, …) ! Seaice
call seq_comm_setcomm(OCNID, …) ! Ocean
call seq_comm_setcomm(CPLID, …) ! Coupler

call seq_comm_joincomm(CPLID,ATMID,CPLATMID) ! Atmoshere and coupler
call seq_comm_joincomm(CPLID,LNDID,CPLLNDID) ! Land and coupler
call seq_comm_joincomm(CPLID,ICEID,CPLICEID) ! Seaice and coupler
call seq_comm_joincomm(CPLID,OCNID,CPLOCNID) ! Ocean and coupler

! Initialize MCT
call mct_world_init(10, MPI_COMM_WORLD, comms, comps)
```

时钟管理



大气

大气化学

陆面

海洋海冰

区域海洋

区域大气

时钟初始化

时间推进

起止时间设置

重启时间设置

时间间隔设置

耦合时间设置

时间步长设置

当前步数设置

输入输出时间设置

闹铃设置

当前步数加1

当前时间向前推进一个步长

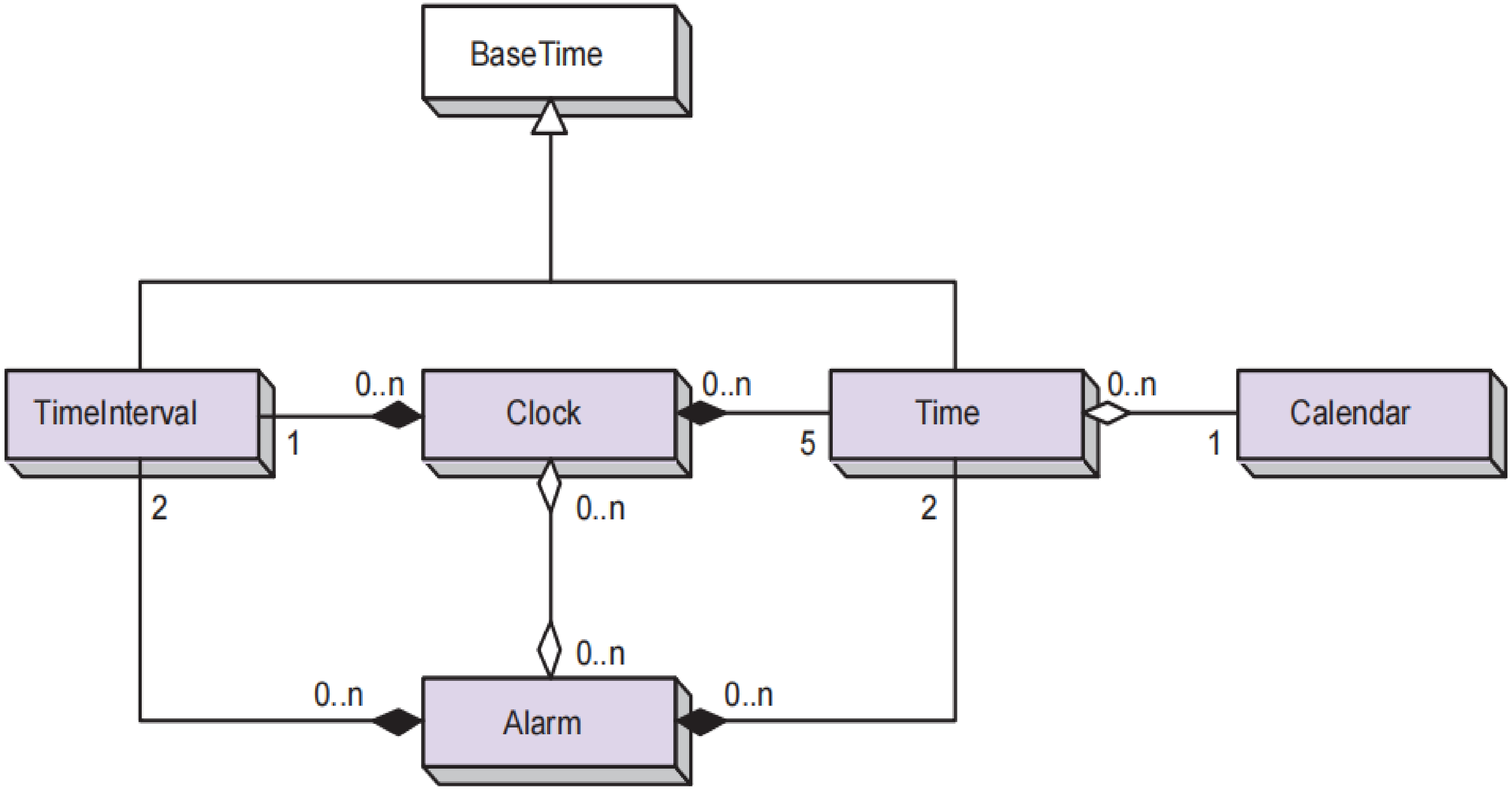
计算下一次闹铃时间

判断闹铃是否响

通知相应模式

Earth Lab

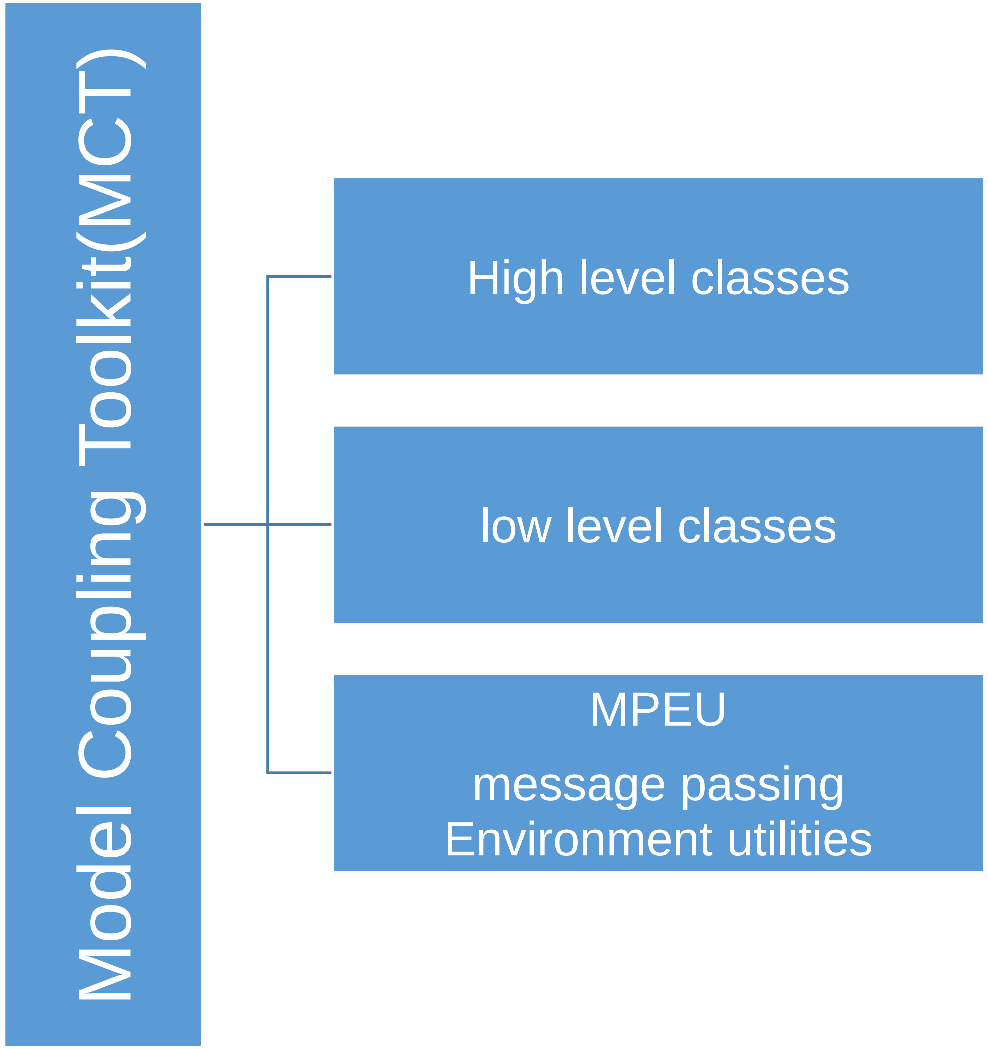
时钟管理


BaseTime
TimeInterval
Clock
Time
Calendar
Alarm
0..n
1
0..n
5
0..n
1
2
0..n
2
0..n
0..n
0..n

时钟管理(models/driver/shr/seq_timemgr_mod.F90)

```
subroutine seq_timemgr_clockAdvance( SyncClock )

   ! --- turn off all alarms on all clocks ---
   do n = 1,max_clocks
      call seq_timemgr_alarmSetOff(SyncClock%ECP(n)%EClock)
   enddo

   ! --- advance driver clock and all driver alarms ---
   call ESMF_ClockAdvance( SyncClock%ECP(seq_timemgr_nclock_drv)%EClock, rc=rc )
  call seq_timemgr_ESMFCodeCheck( rc, msg=subname//"Error from drv ESMF_ClockAdvance")

   if (ESMF_AlarmIsRinging(SyncClock%EAlarm(seq_timemgr_nclock_drv,seq_timemgr_nalarm_atmrun)))
then
      call ESMF_ClockAdvance(SyncClock%ECP(seq_timemgr_nclock_atm)%EClock, rc=rc )
      call seq_timemgr_ESMFCodeCheck(rc, msg=subname//"Error from atm ESMF_ClockAdvance")
   endif
.......

end subroutine seq_timemgr_clockAdvance
```

MCT



Model Coupling Toolkit(MCT)

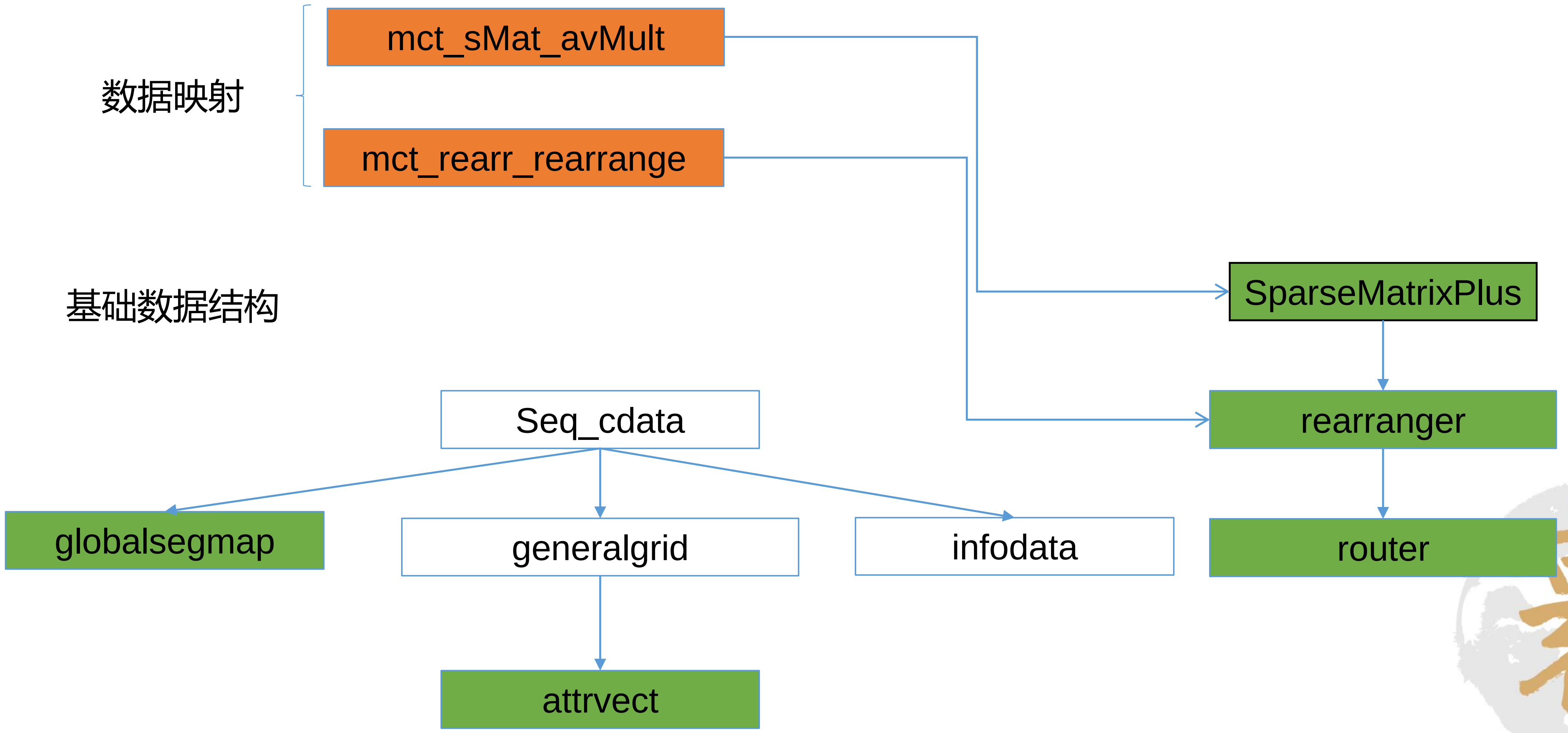
High level classes

low level classes

MPEU

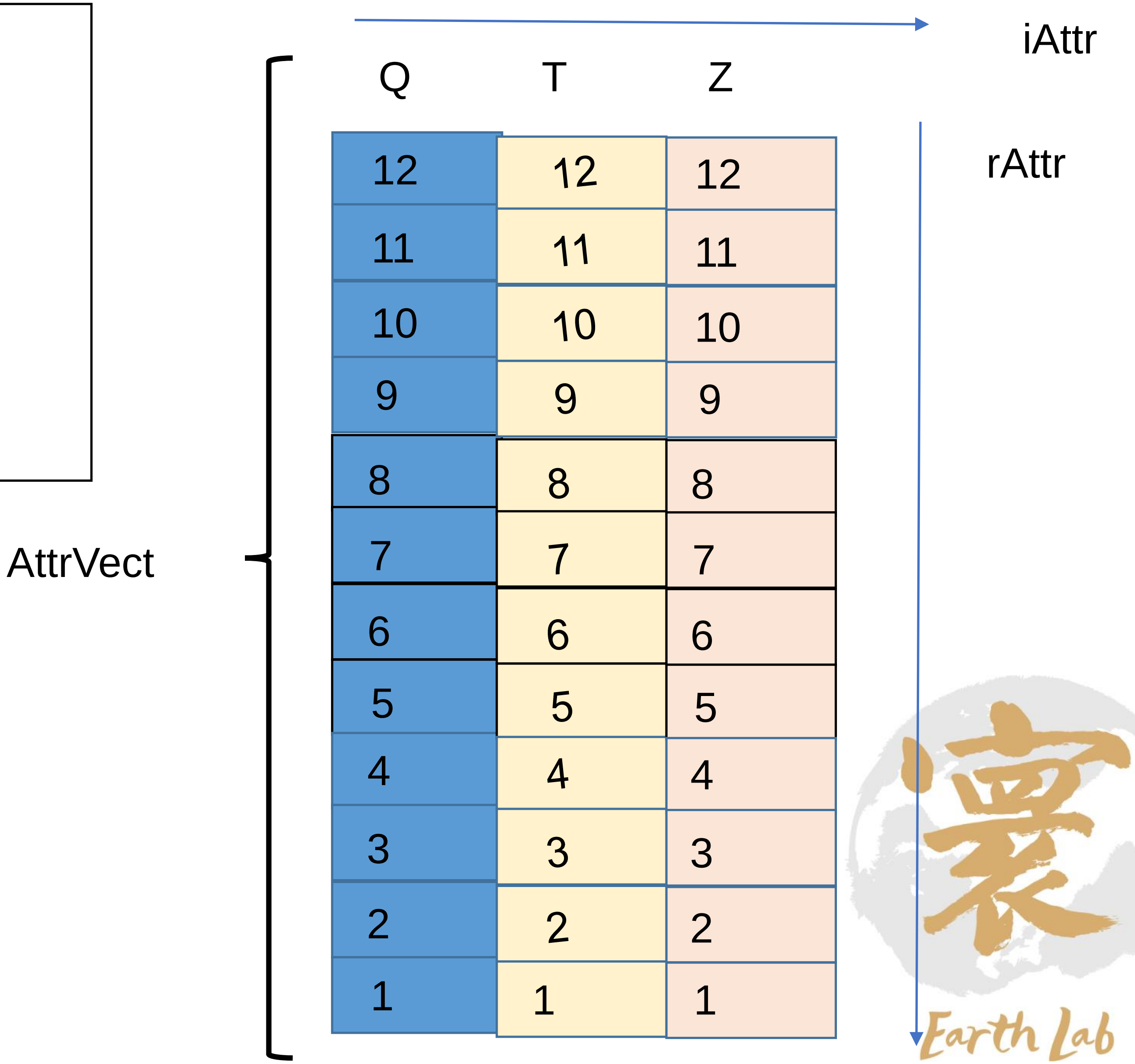
message passing Environment utilities

MCT


数据映射
mct_sMat_avMult
mct_rearr_rearrange
基础数据结构
SparseMatrixPlus
rearranger
router
Seq_cdata
globalsegmap
generalgrid
infodata
attrvect

数据结构

```
    type AttrVect
#ifdef SEQUENCE
      sequence
#endif
      type(List) :: iList
      type(List) :: rList
      integer,dimension(:,:),pointer :: iAttr
      real(FP) ,dimension(:,:),pointer :: rAttr
    end type AttrVec
```



Q	T	Z
12	12	12
11	11	11
10	10	10
9	9	9
8	8	8
7	7	7
6	6	6
5	5	5
4	4	4
3	3	3
2	2	2
1	1	1

数据结构

```
type(mct_aVect) :: x2a_a    ! Atm import, atm grid, atm pes - defined in atm gc
```

```
cam_in(c)%wsx(i)       = -x2a_a%rAttr(index_x2a_Faxx_taux,ig)
   cam_in(c)%wsy(i)       = -x2a_a%rAttr(index_x2a_Faxx_tauy,ig)
   cam_in(c)%lhf(i)       = -x2a_a%rAttr(index_x2a_Faxx_lat, ig)
   cam_in(c)%shf(i)       = -x2a_a%rAttr(index_x2a_Faxx_sen, ig)
   cam_in(c)%lwup(i)      = -x2a_a%rAttr(index_x2a_Faxx_lwup,ig)
   cam_in(c)%cflx(i,1)    = -x2a_a%rAttr(index_x2a_Faxx_evap,ig)
   cam_in(c)%asdir(i)     =  x2a_a%rAttr(index_x2a_Sx_avsdr, ig)
   cam_in(c)%aldir(i)     =  x2a_a%rAttr(index_x2a_Sx_anidr, ig)
   cam_in(c)%asdif(i)     =  x2a_a%rAttr(index_x2a_Sx_avsdf, ig)
   cam_in(c)%aldif(i)     =  x2a_a%rAttr(index_x2a_Sx_anidf, ig)
   cam_in(c)%ts(i)        =  x2a_a%rAttr(index_x2a_Sx_t,     ig)
   cam_in(c)%sst(i)       =  x2a_a%rAttr(index_x2a_So_t,     ig)
   cam_in(c)%snowhland(i) =  x2a_a%rAttr(index_x2a_Sl_snowh, ig)
   cam_in(c)%snowhice(i)  =  x2a_a%rAttr(index_x2a_Si_snowh, ig)
   cam_in(c)%tref(i)      =  x2a_a%rAttr(index_x2a_Sx_tref,  ig)
   cam_in(c)%qref(i)      =  x2a_a%rAttr(index_x2a_Sx_qref,  ig)
   cam_in(c)%u10(i)       =  x2a_a%rAttr(index_x2a_Sx_u10,   ig)
   cam_in(c)%icefrac(i)   =  x2a_a%rAttr(index_x2a_Sa_ifrac, ig)
   cam_in(c)%ocnfrac(i)   =  x2a_a%rAttr(index_x2a_Sa_ofrac, ig)
```

atm_import_mct

数据结构

```
    type GlobalSegMap
 #ifdef SEQUENCE
     sequence
 #endif
     integer :: comp_id                          ! Component ID number
     integer :: ngseg                                    ! No. of Global
segments
     integer :: gsize                                    ! No. of Global
elements
     integer,dimension(:),pointer :: start      ! global seg. start index
     integer,dimension(:),pointer :: length     ! segment lengths
     integer,dimension(:),pointer :: pe_loc     ! PE locations
    end type GlobalSegMap
```

models/atm/cam/src/cpl_mct/atm_comp_mct.F90

atm_SetgsMap_mct

```
 ! Determine global seg map
sizebuf=0
do c = begchunk, endchunk
   ncols = get_ncols_p(c)
   do i = 1,ncols
      sizebuf = sizebuf+1
   end do
end do
allocate(gindex(sizebuf))
n=0
do c = begchunk, endchunk
   ncols = get_ncols_p(c)
   do i = 1,ncols
      n=n+1
      gindex(n) = get_gcol_p(c,i)
   end do
end do

nlcols = get_nlcols_p()
call mct_gsMap_init( gsMap_atm, gindex, mpicom_atm, ATMID, nlcols, ngcols)
```

数据结构

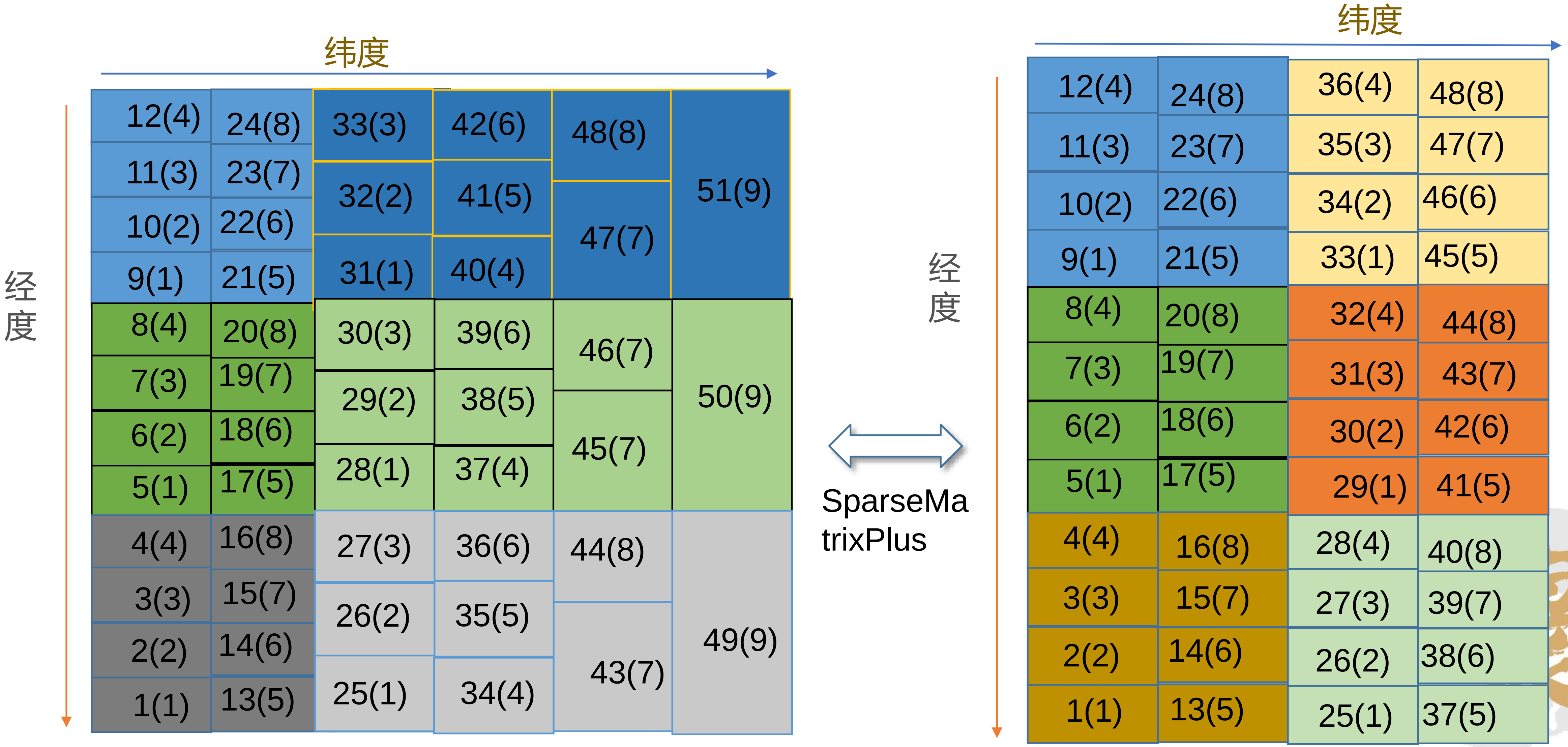
```
type Router
  integer :: comp1id                    ! myid
  integer :: comp2id                    ! id of second component
  integer :: nprocs                          ! number of procs to talk to
  integer :: maxsize                    ! maximum amount of data going to a processor
  integer :: lAvsize                   ! The local size of AttrVect which can be
                                  ! used with this Router in MCT_Send/MCT_Recv
  integer :: numiatt                   ! Number of integer attributes currently in use
  integer :: numratt                   ! Number of real attributes currently in use
  integer,dimension(:),pointer   :: pe_list    ! processor ranks of send/receive in MCT_comm
  integer,dimension(:),pointer   :: num_segs   ! number of segments to send/receive
  integer,dimension(:),pointer   :: locsize    ! total of seg_lengths for a proc
  integer,dimension(:),pointer   :: permarr    ! possible permutation array
  integer,dimension(:,:),pointer :: seg_starts ! starting index
  integer,dimension(:,:),pointer :: seg_lengths! total length
  type(rptr),dimension(:),pointer :: rp1       ! buffer to hold real data
  type(iptr),dimension(:),pointer :: ip1       ! buffer to hold integer data
  integer,dimension(:),pointer   :: ireqs,rreqs  ! buffer for MPI_Requests
  integer,dimension(:,:),pointer :: istatus,rstatus  ! buffer for MPI_Status
end type Router
```

数据结构

```
      type :: Rearranger
#ifdef SEQUENCE
         sequence
#endif
         private
         type(Router) :: SendRouter
         type(Router) :: RecvRouter
         integer,dimension(:,:),pointer :: LocalPack
         integer :: LocalSize
      end type Rearranger
```

```
      Type SparseMatrixPlus
#ifdef SEQUENCE
         sequence
#endif
         type(String) :: Strategy
         integer :: XPrimeLength
         type(Rearranger) :: XToXPrime
         integer :: YPrimeLength
         type(Rearranger) :: YPrimeToY
         type(SparseMatrix) :: Matrix
            integer :: Tag
      End Type SparseMatrixPlus
```

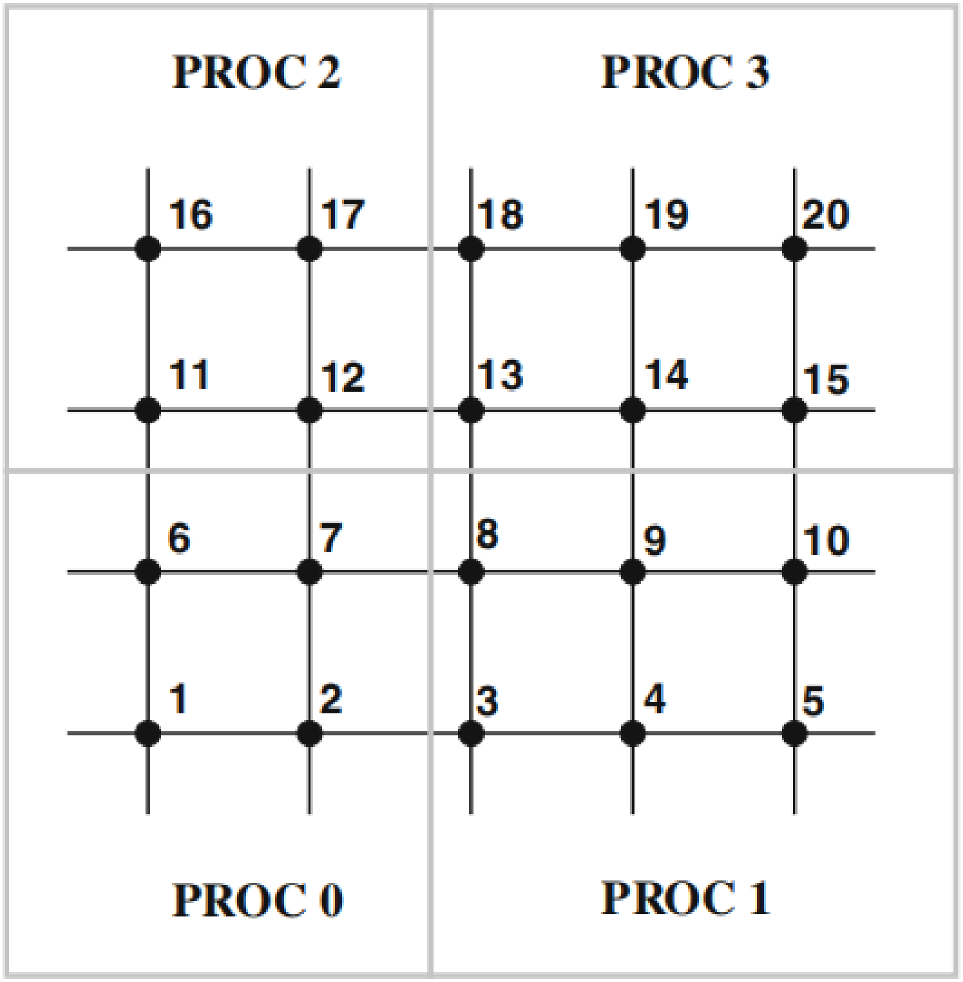
数据结构

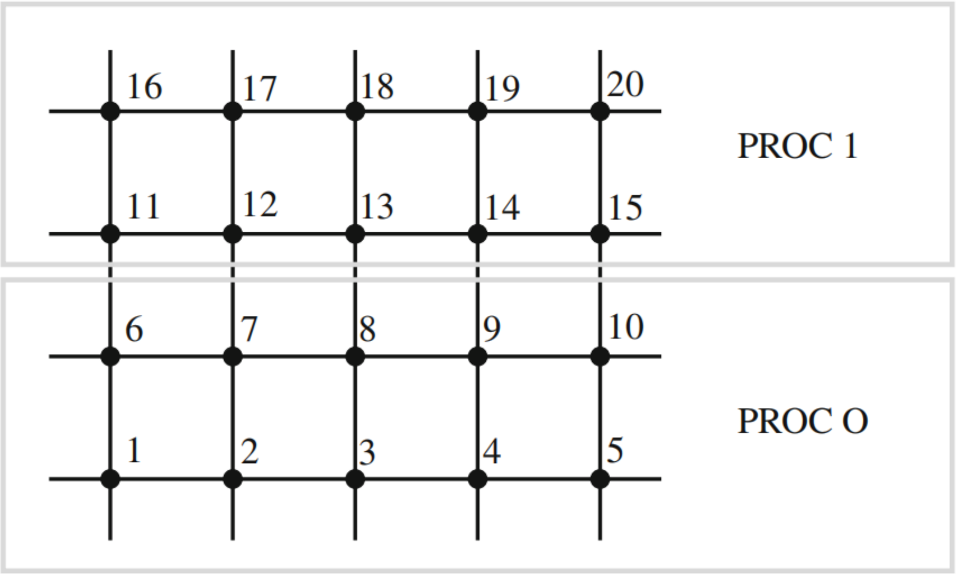


12(4)	24(8)	33(3)	42(6)	48(8)	51(9)
11(3)	23(7)	32(2)	41(5)	47(7)	
10(2)	22(6)	31(1)	40(4)		
9(1)	21(5)				
8(4)	20(8)	30(3)	39(6)	46(7)	50(9)
7(3)	19(7)	29(2)	38(5)	45(7)	
6(2)	18(6)	28(1)	37(4)		
5(1)	17(5)				
4(4)	16(8)	27(3)	36(6)	44(8)	49(9)
3(3)	15(7)	26(2)	35(5)	43(7)	
2(2)	14(6)	25(1)	34(4)		
1(1)	13(5)				

12(4)	24(8)	36(4)	48(8)
11(3)	23(7)	35(3)	47(7)
10(2)	22(6)	34(2)	46(6)
9(1)	21(5)	33(1)	45(5)
8(4)	20(8)	32(4)	44(8)
7(3)	19(7)	31(3)	43(7)
6(2)	18(6)	30(2)	42(6)
5(1)	17(5)	29(1)	41(5)
4(4)	16(8)	28(4)	40(8)
3(3)	15(7)	27(3)	39(7)
2(2)	14(6)	26(2)	38(6)
1(1)	13(5)	25(1)	37(5)

Rearrange





Rearranger

网格映射

- 算法
  - Y=AX，模式X到模式Y，A是对角线稀疏矩阵
  - 双线性插值，不守恒，但很好保持了分布和梯度，适用于标量
  - 二阶面积守恒，通量守恒，在一定程度上保持形状和梯度
  - PATCH（面积守恒），通量守恒
- 软件
  - SCRIPS，Fortran
  - ESMF，python或者NCL
  - tempestremap

SCRIP

```
COMPILE = gfortran
FLAGS = -L/cygdrive/c/netcdf-3.6.2/lib -I/cygdrive/c/netcdf-3.6.2/include
#FLAGS = -L/usr/lib -I/usr/include
LIB  =  -lnetcdf
```

```
make
./scrip  scrip_in
```

```
&remap_inputs
    num_maps = 2
    grid1_file = '../grid-jra55.nc'
    grid2_file = '../grid-30m.nc'
    interp_file1 = 'jra55_to_30m_conserv.nc'
    interp_file2 = '30m_to_jra55_conserv.nc'
    map1_name = 'JRA55 to 30m Conservative Mapping'
    map2_name = '30m to JRA55 Conservative Mapping'
    map_method = 'conservative'
    normalize_opt = 'frac'
    output_opt = 'scrip'
    restrict_type = 'latitude'
    num_srch_bins = 90
    luse_grid1_area = .false.
    luse_grid2_area = .false.
/
```

SCRIP的格点文件

- create-camgrid.F90, create-ocngrid.F90, create-wrfgrid.f90, create-geatmgrid.f90, create-lndgrid.f90, create-romsgrid.f90
- netcdf cam-global.grid {

```
    dimensions:
 grid_size = 32768 ;
 grid_corners = 4 ;
 grid_rank = 2 ;
  variables:
 int grid_dims(grid_rank) ;
 double grid_center_lat(grid_size) ;
           grid_center_lat:units = "degrees" ;
 double grid_center_lon(grid_size) ;
           grid_center_lon:units = "degrees" ;
 int grid_imask(grid_size) ;
           grid_imask:units = "unitless" ;
 double grid_area(grid_size) ;
           grid_area:radiant = "unitless" ;
 double grid_corner_lat(grid_size, grid_corners) ;
           grid_corner_lat:units = "degrees" ;
 double grid_corner_lon(grid_size, grid_corners) ;
           grid_corner_lon:units = "degrees" ;
```

SCRIP的映射文件

```
netcdf licom_to_IAPGCM_aave_da_20140402 {
dimensions:
	n_a = 70560 ;
	n_b = 32768 ;
	nv_a = 4 ;
	nv_b = 4 ;
	src_grid_rank = 2 ;
	dst_grid_rank = 2 ;
	ni_a = 360 ;
	nj_a = 196 ;
	ni_b = 256 ;
	nj_b = 128 ;
	n_s = 132433 ;
	num_wgts = 3 ;
	num_wgts1 = 2 ;
variables:
	int src_grid_dims(src_grid_rank) ;
	int dst_grid_dims(dst_grid_rank) ;
	double yc_a(n_a) ;
		yc_a:units = "degrees" ;
	double yc_b(n_b) ;
		yc_b:units = "degrees" ;
	double xc_a(n_a) ;
		xc_a:units = "degrees" ;
	double xc_b(n_b) ;
		xc_b:units = "degrees" ;
	double yv_a(n_a, nv_a) ;
		yv_a:units = "degrees" ;
	double xv_a(n_a, nv_a) ;
		xv_a:units = "degrees" ;
	double yv_b(n_b, nv_b) ;
		yv_b:units = "degrees" ;
	double xv_b(n_b, nv_b) ;
		xv_b:units = "degrees" ;
	int mask_a(n_a) ;
		mask_a:units = "unitless" ;
	int mask_b(n_b) ;
		mask_b:units = "unitless" ;
	double area_a(n_a) ;
		area_a:units = "square radians" ;
	double area_b(n_b) ;
		area_b:units = "square radians" ;
	double frac_a(n_a) ;
		frac_a:units = "unitless" ;
	double frac_b(n_b) ;
		frac_b:units = "unitless" ;
	int col(n_s) ;
	int row(n_s) ;
	double S(n_s) ;
	double S2(n_s, num_wgts1) ;
```

SCRIP的映射文件

```
netcdf licom_to_IAPGCM_aave_da_20140402 {
dimensions:
        n_a = 70560 ;
        n_b = 32768 ;
        nv_a = 4 ;
        nv_b = 4 ;
        src_grid_rank = 2 ;
        dst_grid_rank = 2 ;
        ni_a = 360 ;
        nj_a = 196 ;
        ni_b = 256 ;
        nj_b = 128 ;
        n_s = 132433 ;
        num_wgts = 3 ;
        num_wgts1 = 2 ;
variables:
        int col(n_s) ;
        int row(n_s) ;
        double S(n_s) ;
        double S2(n_s, num_wgts1) ;
```

```
do n=1,n_s
   dst_array(row(n)) = dst_array(row(n)) + &
                 src_array(col(n))*S2(n,1) + &
                 src_array_grad1(col(n))*S2(n,2) + &
                 src_array_grad2(col(n))*S2(n,3)
end do
```

NCL

```
;----------------------------------------------------------------------
; ncep-iap
;----------------------------------------------------------------------
    srcGridName = "src_iap_ncep.nc"
    dstGridName = "dst_iap_ncep.nc"
    wgtFileName = "weight_iap_ncep.nc"

;----------------------------------------------------------------------
; Convert original source IAP grid to a SCRIP convention file.
;----------------------------------------------------------------------
    Opt                   = True
    Opt@ForceOverwrite = True
    Opt@Title          = "IAP Grid"

    b = addfile("/data/hjx/RUN_MODEL/dataoutput/2020/20200301/ENS01/huanan-1x1test.cam2.h1.2020-06-10-00000.nc","r")
    rectilinear_to_SCRIP(srcGridName,b->lat,b->lon,Opt)
    delete(Opt)
```

NCL

```
;----------------------------------------------------------------------
; Convert destination grid to a SCRIP convention file.
;----------------------------------------------------------------------
    Opt                = True
    Opt@ForceOverwrite = True

    netcdf_name="/data/hjx/RUN_MODEL/climatology-data/air.2016.nc"  ; from NCC
    a = addfile( netcdf_name, "r")
    lat=a->lat
    lon=a->lon
    rectilinear_to_SCRIP(dstGridName,a->lat,a->lon,Opt)
    delete(Opt)
;----------------------------------------------------------------------
; Generate the weights
;----------------------------------------------------------------------
    Opt                      = True
    Opt@InterpMethod         = "bilinear"     ; default
    Opt@ForceOverwrite       = True
    Opt@DstRegional          = True
    Opt@SrcRegional          = True

    ESMF_regrid_gen_weights(srcGridName,dstGridName,wgtFileName,Opt)
```

xEMSF

- conda install -c conda-forge xesmf esmpy=7.1.0

```
import xesmf as xe
from netCDF4 import Dataset

fnc0 = Dataset('/mnt/i/era5-monthly/era5-t850.monthly.nc')
fnc20 = Dataset('/mnt/i/30years-monmean/old-iap-t.monmean.nc')
late=fnc0.variables["latitude"][:].astype(float)
lone=fnc0.variables["longitude"][:].astype(float)
lat=fnc20.variables["lat"][:].astype(float)
lon=fnc20.variables["lon"][:].astype(float)

grid_in = {'lon': lon, 'lat': lat}
grid_out = {'lon': lone, 'lat': late}
regridder = xe.Regridder(grid_in, grid_out, 'patch')
regridder.clean_weight_file()
regridder

iotv_regrid = regridder(iotv)  # regrid a basic numpy array
```

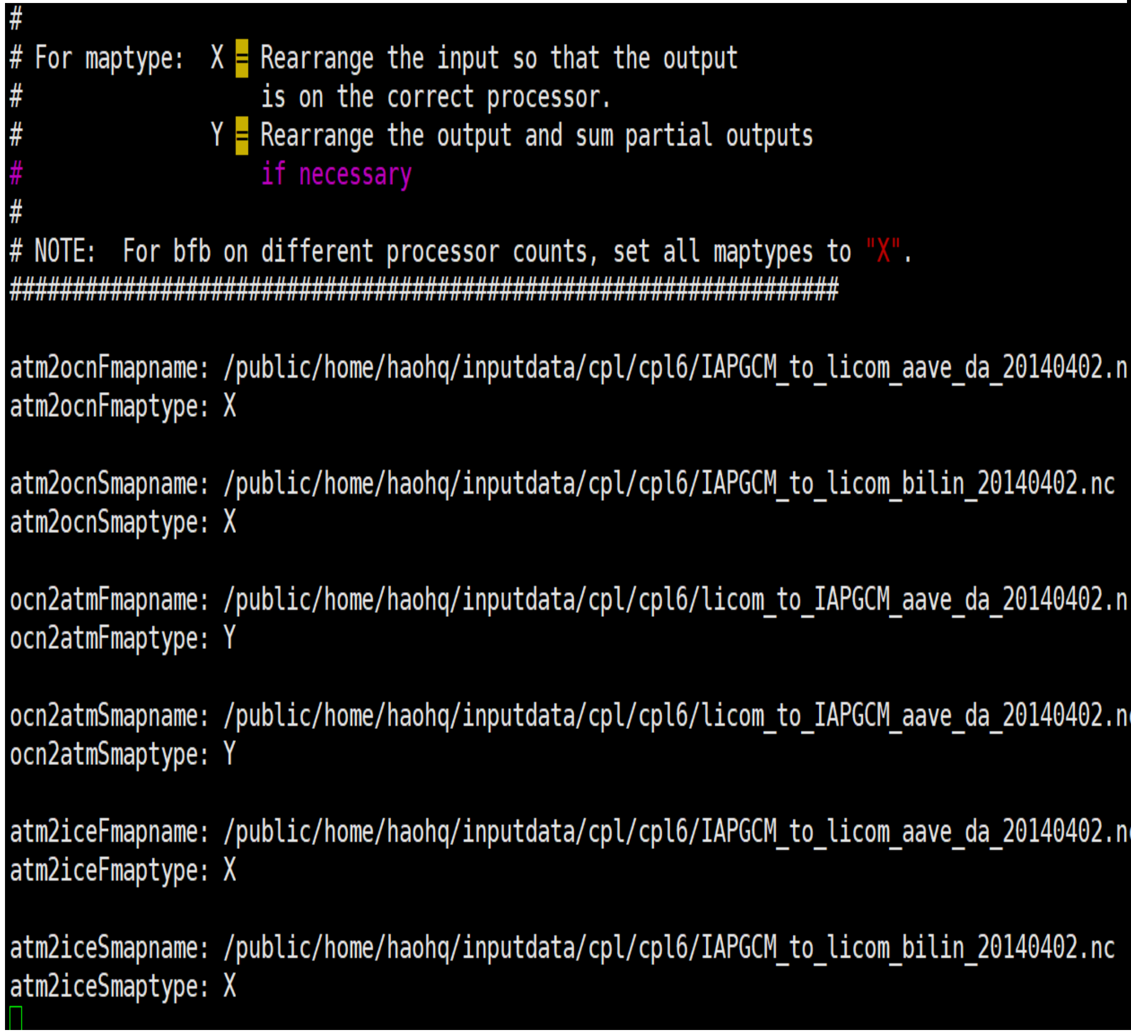
tempest-remap

-
  ```
  conda install -c conda-forge tempest-remap
  ```
- 有限元，有限体元，谱元等

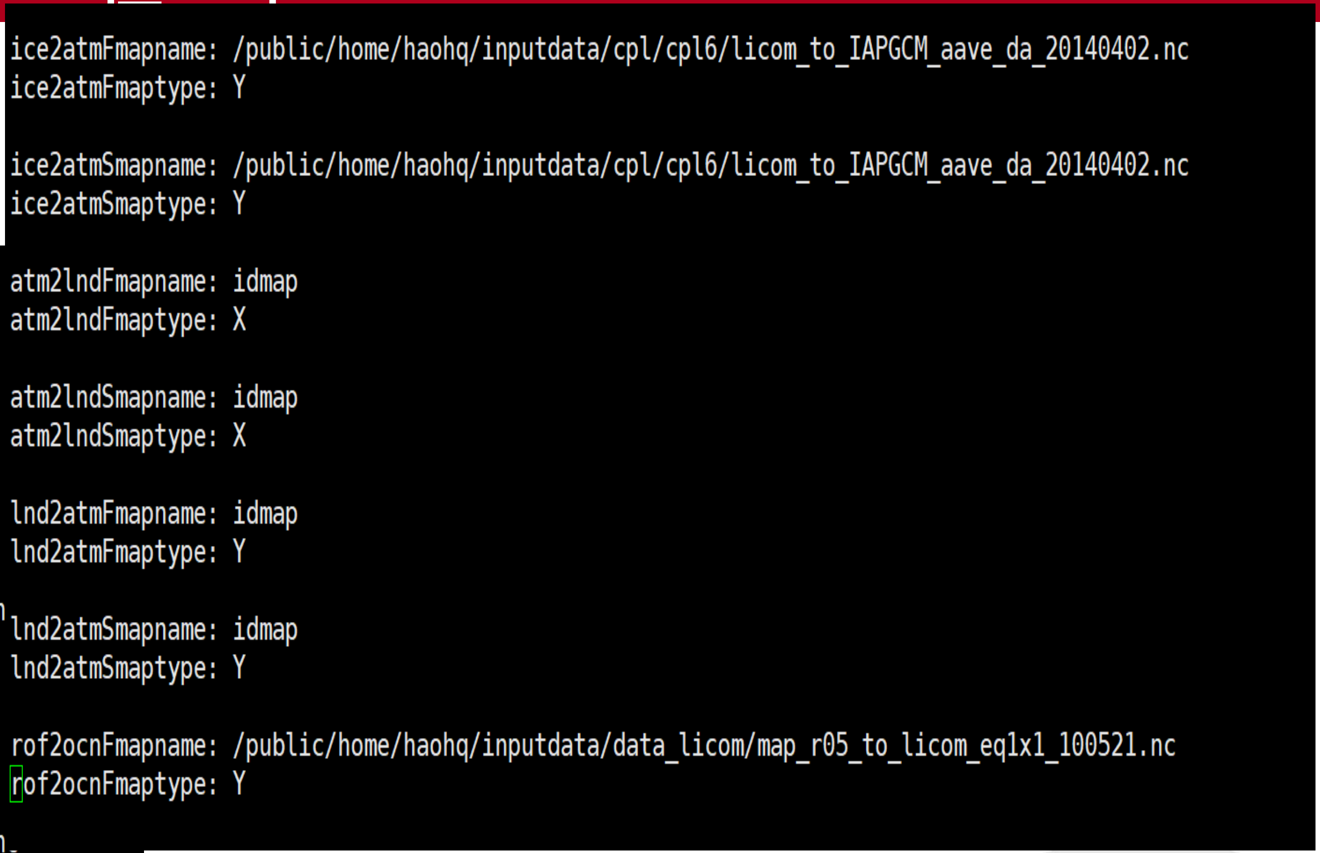
-
  ```
  ./GenerateRLLMesh --lon <longitudes> --lat <latitudes> --file <Output mesh filename>.g
  ```

-
  ```
  ./GenerateOfflineMap --in_mesh <Input mesh>.g --out_mesh <Output mesh>.g \
                --ov_mesh <Overlap mesh>.g --in_np <Remapping Order> \
                  --out_map <Output map>.nc
  ```

seq_maps.rc

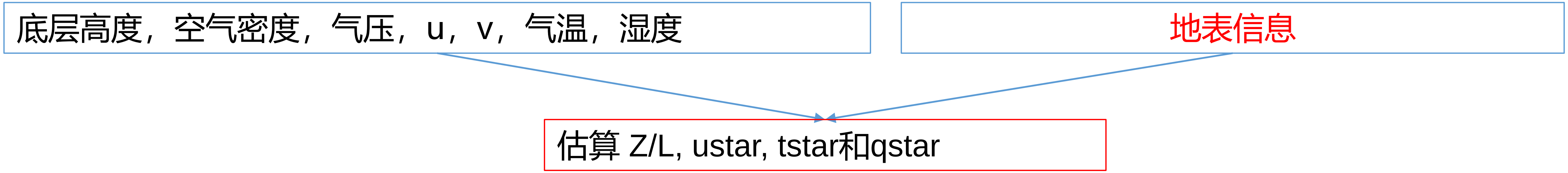


```
#
# For maptype:  X = Rearrange the input so that the output
#                   is on the correct processor.
#               Y = Rearrange the output and sum partial outputs
#                   if necessary
#
# NOTE:  For bfb on different processor counts, set all maptypes to "X".
##################################################################

atm2ocnFmapname: /public/home/haohq/inputdata/cpl/cpl6/IAPGCM_to_licom_aave_da_20140402.n
atm2ocnFmaptype: X

atm2ocnSmapname: /public/home/haohq/inputdata/cpl/cpl6/IAPGCM_to_licom_bilin_20140402.nc
atm2ocnSmaptype: X

ocn2atmFmapname: /public/home/haohq/inputdata/cpl/cpl6/licom_to_IAPGCM_aave_da_20140402.n
ocn2atmFmaptype: Y

ocn2atmSmapname: /public/home/haohq/inputdata/cpl/cpl6/licom_to_IAPGCM_aave_da_20140402.nc
ocn2atmSmaptype: Y

atm2iceFmapname: /public/home/haohq/inputdata/cpl/cpl6/IAPGCM_to_licom_aave_da_20140402.nc
atm2iceFmaptype: X

atm2iceSmapname: /public/home/haohq/inputdata/cpl/cpl6/IAPGCM_to_licom_bilin_20140402.nc
atm2iceSmaptype: X
```



```
ice2atmFmapname: /public/home/haohq/inputdata/cpl/cpl6/licom_to_IAPGCM_aave_da_20140402.nc
ice2atmFmaptype: Y

ice2atmSmapname: /public/home/haohq/inputdata/cpl/cpl6/licom_to_IAPGCM_aave_da_20140402.nc
ice2atmSmaptype: Y

atm2lndFmapname: idmap
atm2lndFmaptype: X

atm2lndSmapname: idmap
atm2lndSmaptype: X

lnd2atmFmapname: idmap
lnd2atmFmaptype: Y

lnd2atmSmapname: idmap
lnd2atmSmaptype: Y

rof2ocnFmapname: /public/home/haohq/inputdata/data_licom/map_r05_to_licom_eq1x1_100521.nc
rof2ocnFmaptype: Y
```

通量计算


底层高度，空气密度，气压，u，v，气温，湿度

地表信息

估算 Z/L, ustar, tstar和qstar

$$\vec{\tau}_a = \frac{\rho_a u^{*2}\vec{U}_a}{\left|\vec{U}_a\right|}$$

$$F_s = C_s(\Theta_a - T_{sf}^K)$$

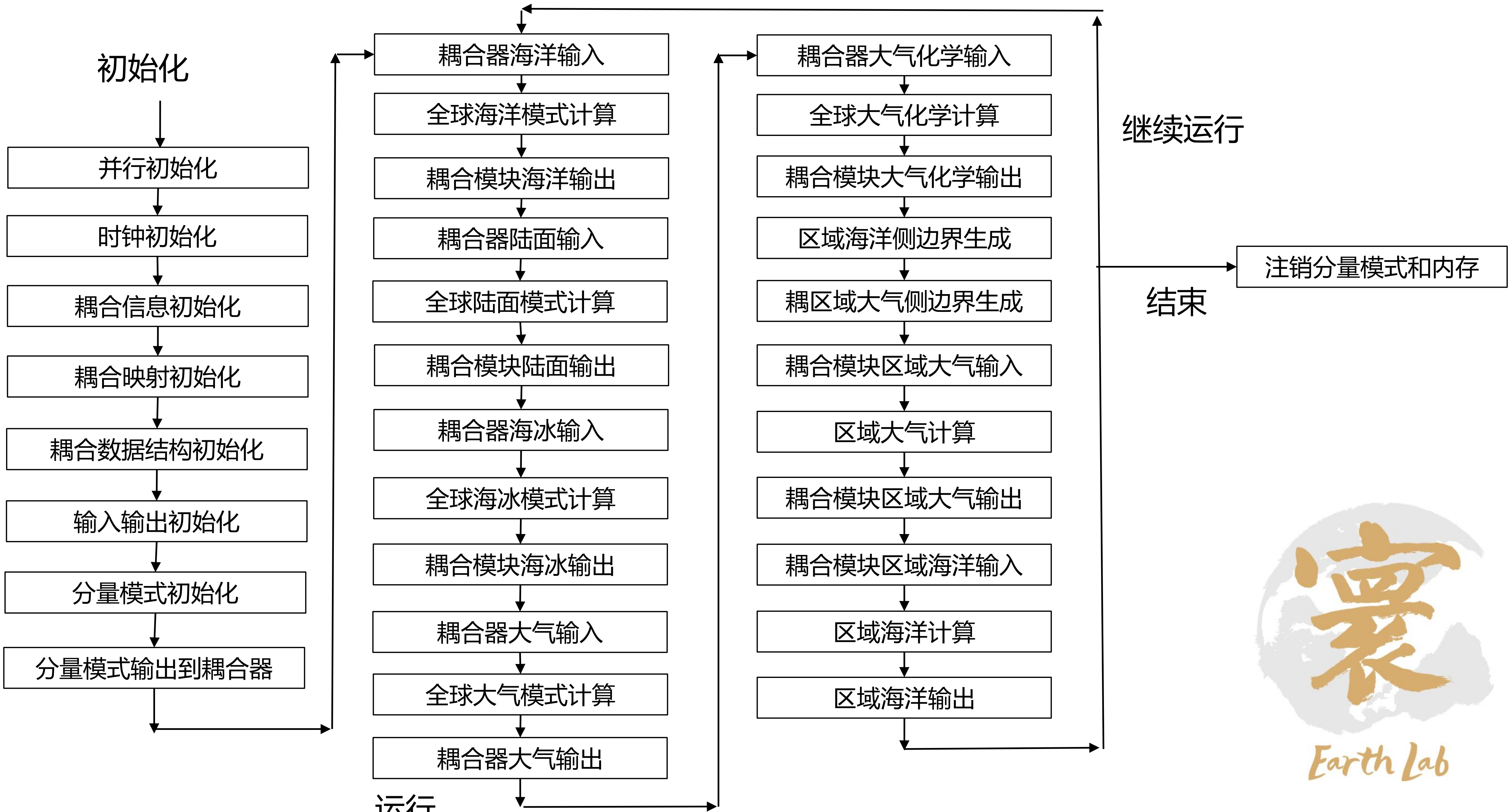
$$F_l = C_l(\Theta_a - Q_{sf})$$

1. compute stability & evaluate all stability functions、
2. shift wind speed using old coefficient
3. update transfer coeffs at 10m and neutral stability
4. shift all coeffs to measurement height and stability
5. update ustar, tstar, qstar using updated, shifted coeffs
6. update transfer coeffs at 10m and neutral stability
7. shift all coeffs to measurement height and stability
8. update ustar, tstar, qstar using updated, shifted coeffs
9. compute the fluxes
10. iterate to converge on Z/L, ustar, tstar and qstar

通量计算

- 海气通量计算在耦合器中进行
  - csm_share/shr/shr_flux_mod.F90
- 陆气通量在陆面模式中进行
  - 次网格计算后汇集，colm/src/mainc/fluxave.F90
  - colm/src/mainc/moninobuk.F90
- 冰气通量在海冰模式中进行
  - 朝向大气表面，cice/src/icepack/columnphysics/icepack_atmo.F90
  - 朝向海洋表面，cice/src/icepack/columnphysics/icepack_ocean.F90

总控程序



初始化

并行初始化

时钟初始化

耦合信息初始化

耦合映射初始化

耦合数据结构初始化

输入输出初始化

分量模式初始化

分量模式输出到耦合器

耦合器海洋输入

全球海洋模式计算

耦合模块海洋输出

耦合器陆面输入

全球陆面模式计算

耦合模块陆面输出

耦合器海冰输入

全球海冰模式计算

耦合模块海冰输出

耦合器大气输入

全球大气模式计算

耦合器大气输出

运行

耦合器大气化学输入

全球大气化学计算

耦合模块大气化学输出

区域海洋侧边界生成

耦区域大气侧边界生成

耦合模块区域大气输入

区域大气计算

耦合模块区域大气输出

耦合模块区域海洋输入

区域海洋计算

区域海洋输出

继续运行

结束

注销分量模式和内存

各分量模式*_comp_*.F

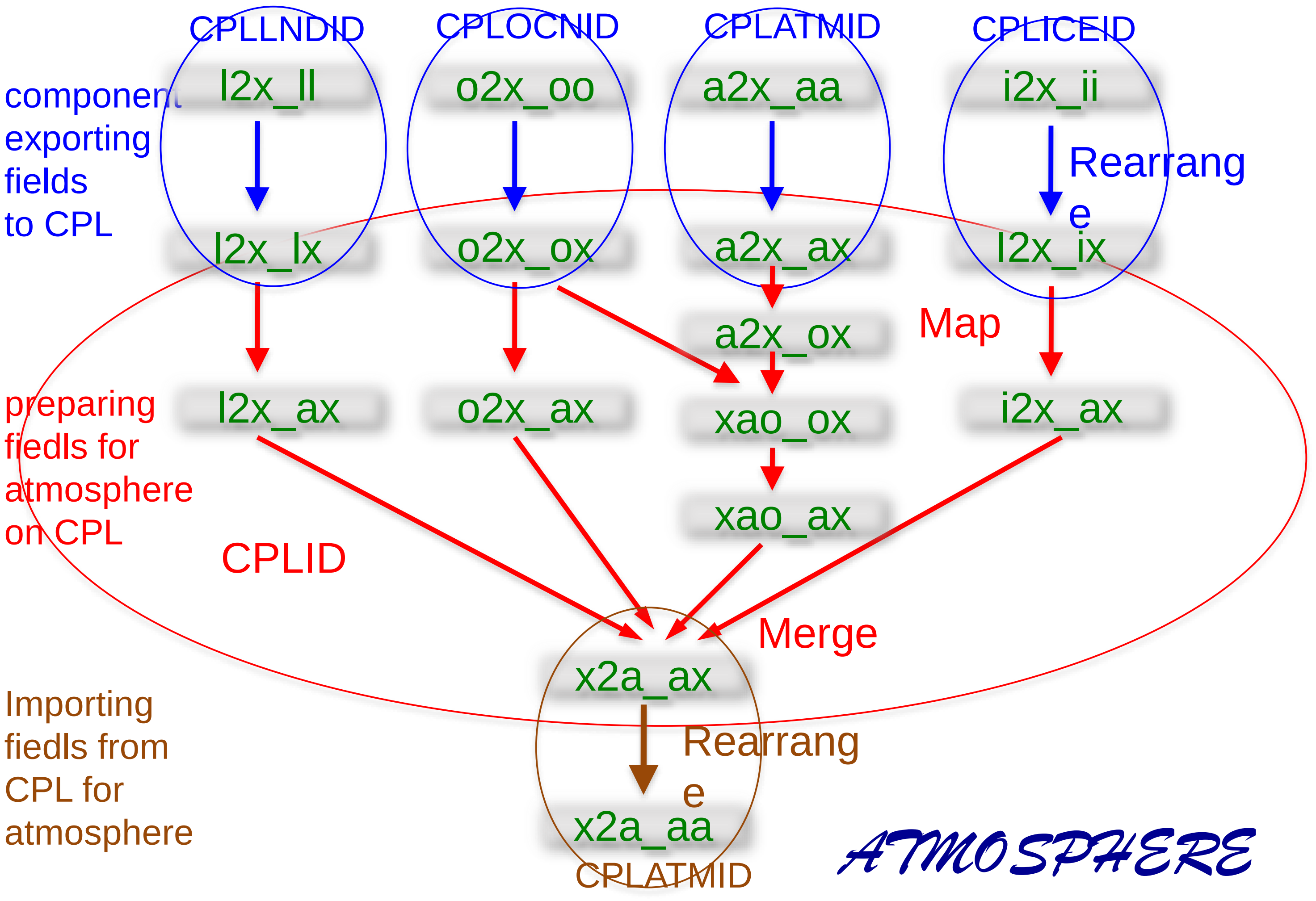
- 公共接口：
  - Init, run, final
- 内部接口
  - Segtgsmap, domain格点分布和映射
  - Import, export在积分前输入，积分后输出
  - Read_srfrest, write_srfrest重启文件

各分量模式*_comp_*.F

```
module atm_comp_mct
  public :: atm_init_mct
  public :: atm_run_mct
  public :: atm_final_mct
  private :: atm_SetgsMap_mct
  private :: atm_import_mct
  private :: atm_export_mct
  private :: atm_domain_mct
  private :: atm_read_srfrest_mct
  private :: atm_write_srfrest_mct
  type(cam_in_t) , pointer :: cam_in(:)
  type(cam_out_t), pointer :: cam_out(:)
```

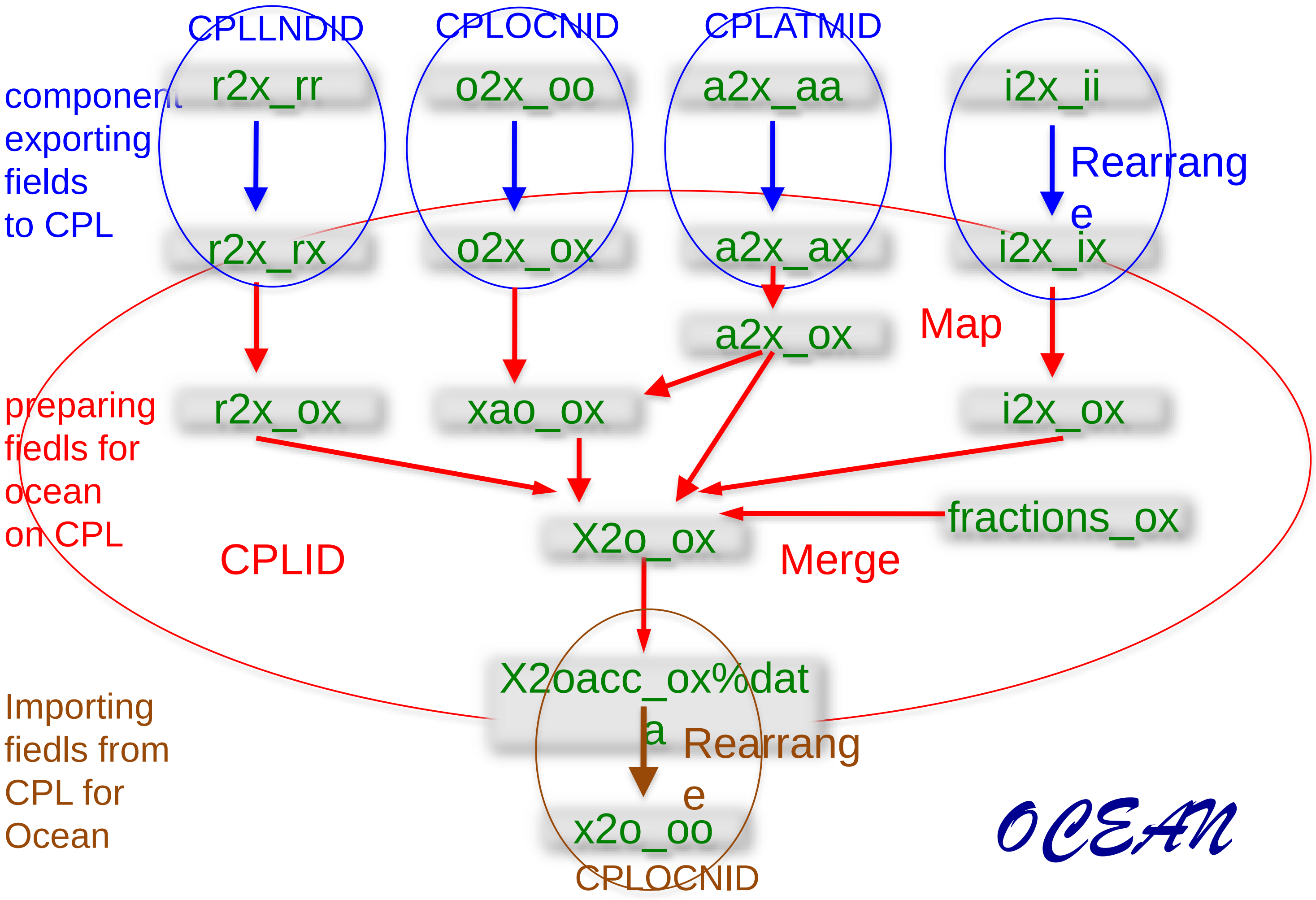
```
module ocn_comp_mct
 public :: ocn_init_mct
  public :: ocn_run_mct
  public :: ocn_final_mct
private :: ocn_export_mct
  private :: ocn_import_mct
  private :: ocn_SetGSMap_mct
  private :: ocn_domain_mct
```

大气接收数据



ATMOSPHERE

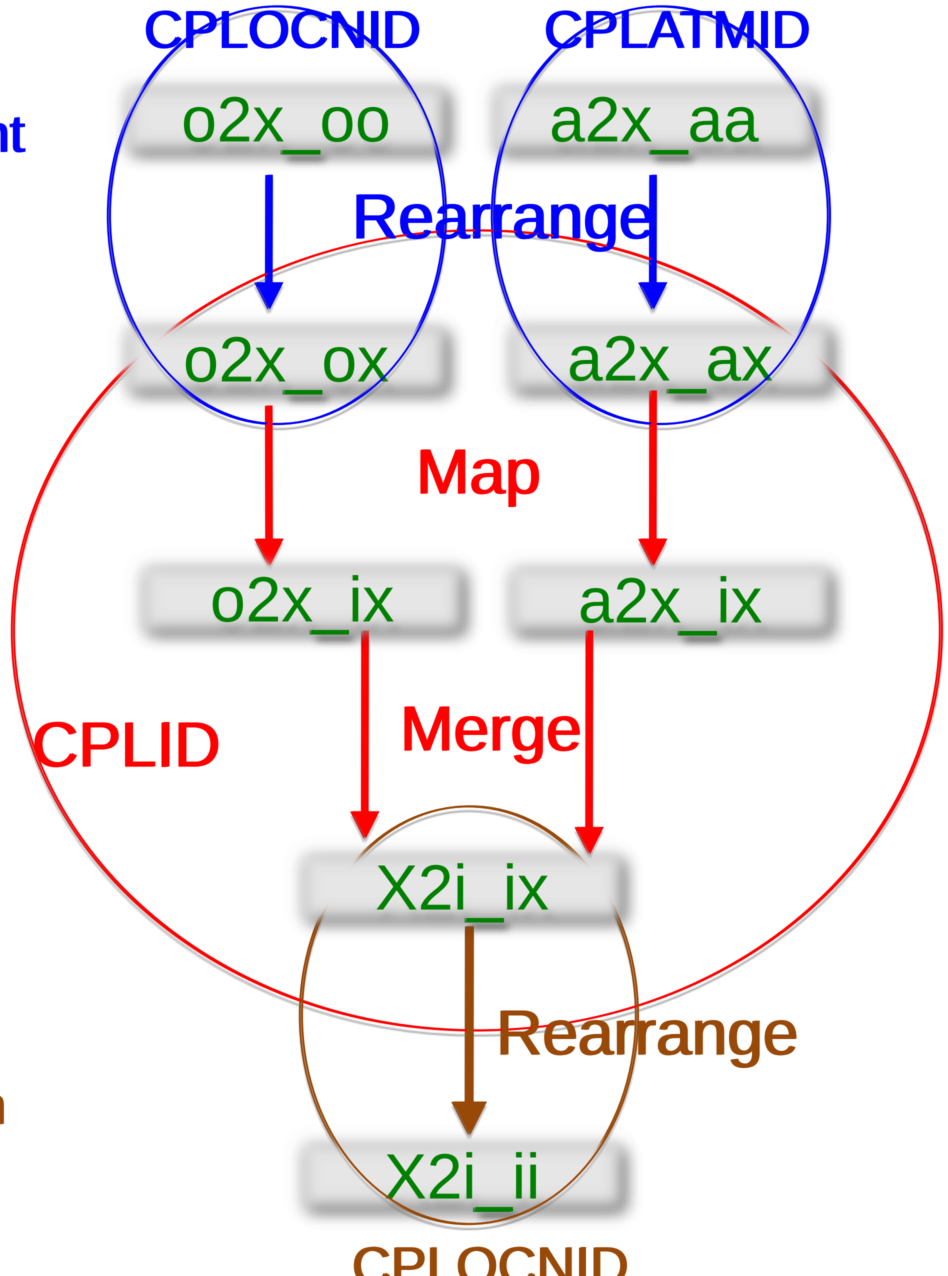
海洋接收数据



OCEAN

Earth Lab

陆面接收数据



component exporting fields to CPL

preparing fiedls for seaice on CPL

Importing fiedls from CPL for seaice

LAND

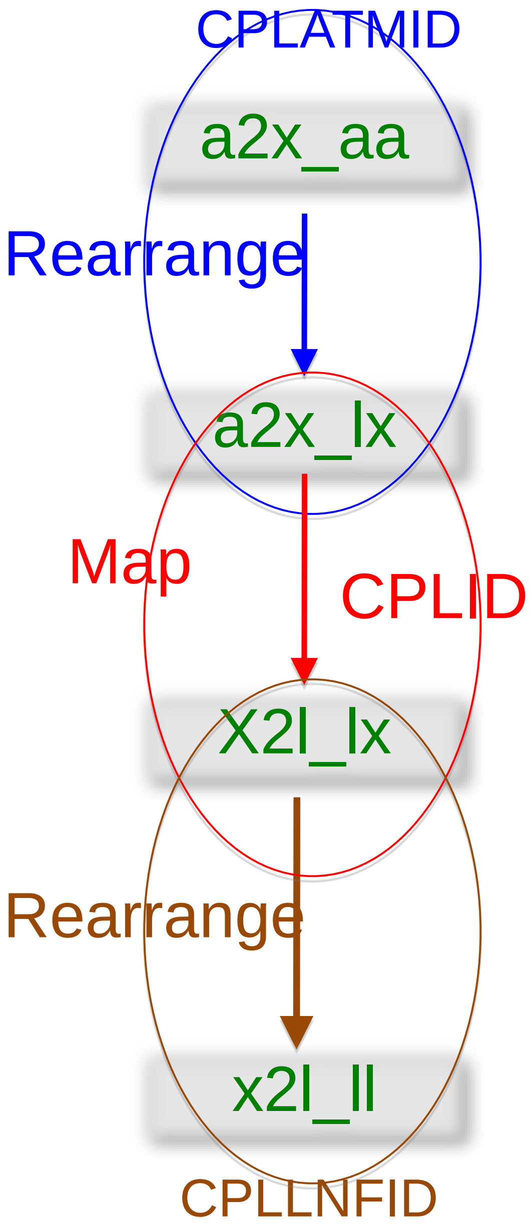
Earth Lab

海冰接收数据

component
exporting
fields
to CPL

preparing
fiedls for
seaice
on CPL

Importing
fiedls from
CPL for
seaice



CPLATMID

a2x_aa

Rearrange

a2x_lx

Map

CPLID

X2l_lx

Rearrange

x2l_ll

CPLLNFID

SEAICE

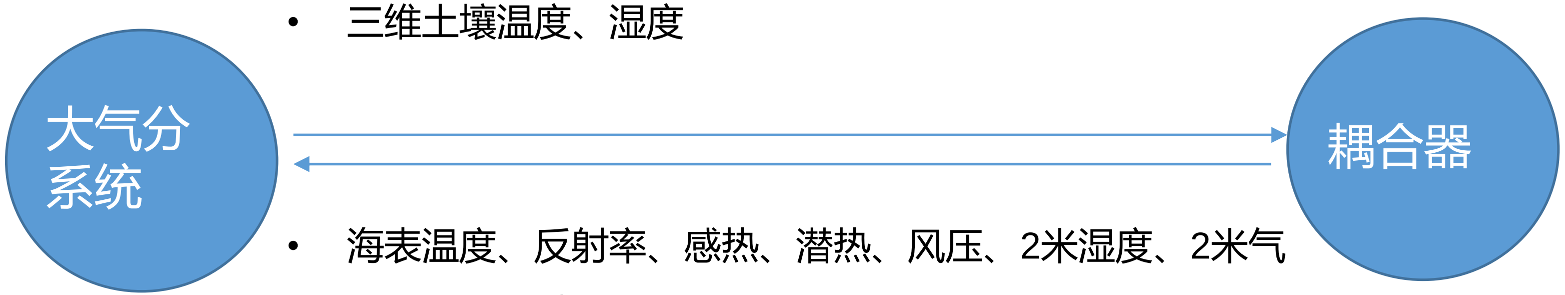
Earth Lab

大气模式收发数据和运行

```
if (iamin_CPLID .and. atm_prognostic) then
        call map_ocn2atm_mct  ! o2x_ox ->o2x_ax, xao_ox->xao_ax
        call map_ice2atm_mct  !l2x_ix ->i2x_ax
        call map_lnd2atm_mct  !l2x_lx ->l2x_ax
        call mrg_x2a_run_mct  ! l2x_ax+o2x_ax+xao_ax+i2x_ax+fractions_ax->x2a_ax
    endif

    if (iamin_CPLATMID .and. atm_prognostic) then
      call map_atmx2atma_mct   ! x2a_ax -> x2a_aa
    endif                    ! Rearrange and import

    call atm_run_mct( EClock_a, cdata_aa, x2a_aa, a2x_aa)  !export a2x_aa

    if (iamin_CPLATMID) then
      call map_atma2atmx_mct   !a2x_aa->a2x_ax  ! Rearrange and export
    endif
 if (iamin_CPLID) then
call mct_avect_vecmult(a2x_ax,mdl2drv_ax,seq_flds_a2x_fluxes) !acumulate
endif
```

大气收发数据

- 三维气溶胶，如沙尘、海盐、黑炭、硫酸盐、硝酸盐等
- 降水、降雪、长短波辐射
- 碳、氮、磷通量
- 三维风、温、压、湿、密度、位势高度
- 三维土壤温度、湿度

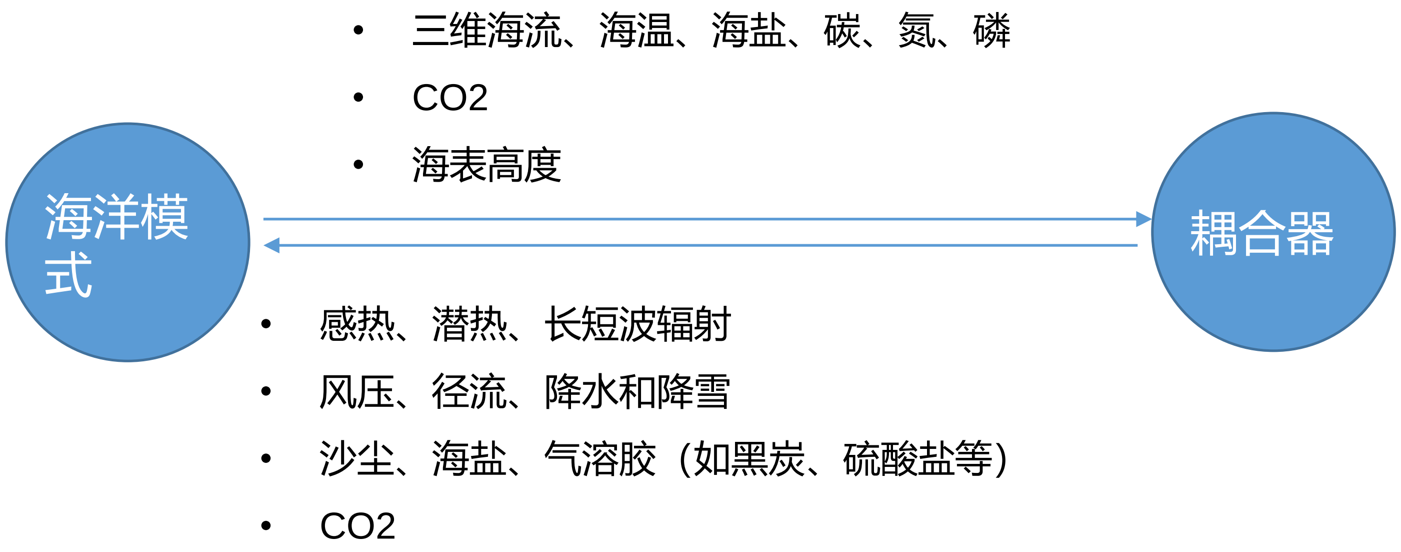


大气分系统

耦合器

- 海表温度、反射率、感热、潜热、风压、2米湿度、2米气温、10米风速
- 积雪
- 碳、氮、磷排放
- 三维风、温、压、湿、密度、位势高度
- 三维风、温、压、湿、密度、位势高度的趋势项
- 三维气溶胶排放，如沙尘、黑炭、硫酸盐、硝酸盐等

海洋收发数据



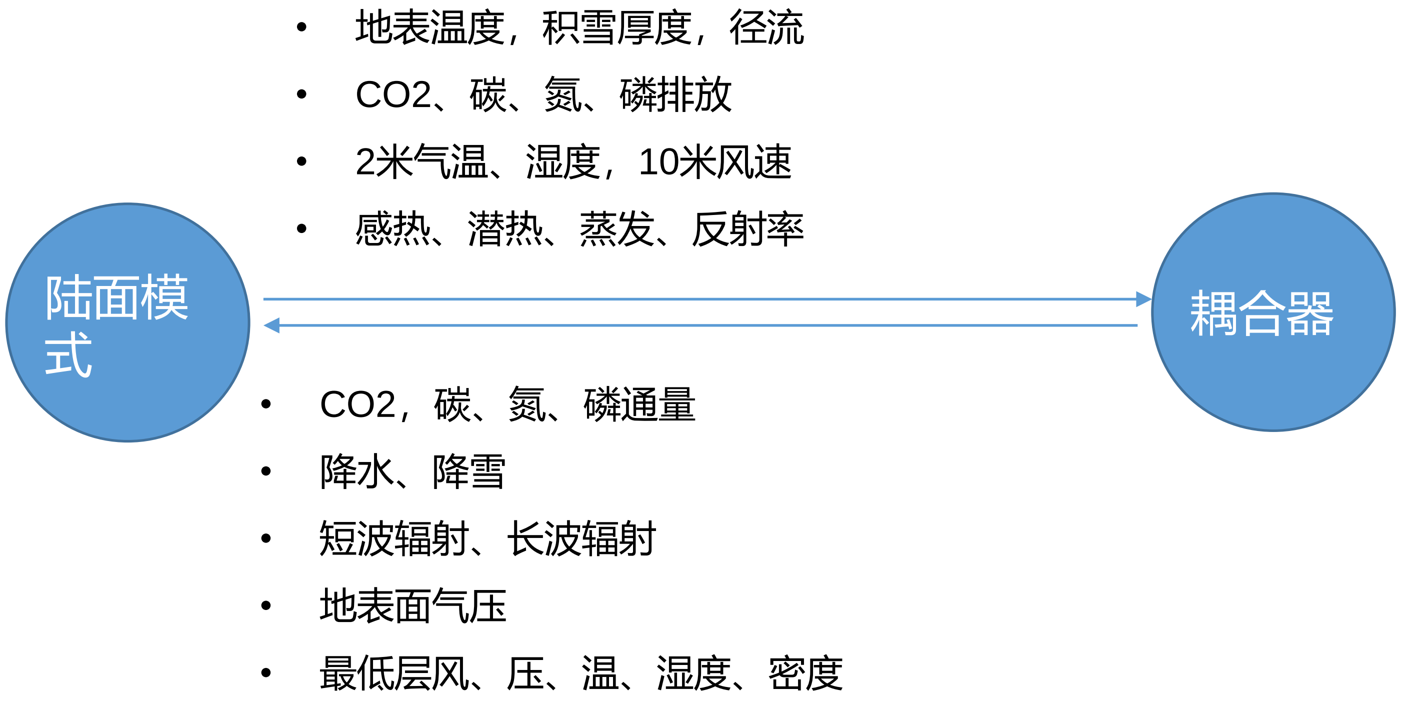
海洋模式

耦合器

- 三维海流、海温、海盐、碳、氮、磷
- CO2
- 海表高度

- 感热、潜热、长短波辐射
- 风压、径流、降水和降雪
- 沙尘、海盐、气溶胶（如黑炭、硫酸盐等）
- CO2

陆面收发数据



- 地表温度，积雪厚度，径流
- CO2、碳、氮、磷排放
- 2米气温、湿度，10米风速
- 感热、潜热、蒸发、反射率

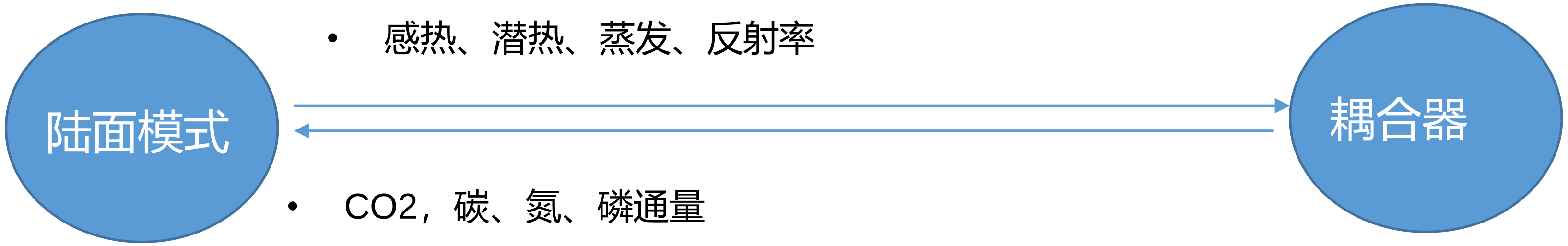
陆面模式

耦合器

- CO2，碳、氮、磷通量
- 降水、降雪
- 短波辐射、长波辐射
- 地表面气压
- 最低层风、压、温、湿度、密度

陆面收发数据

- 地表温度，积雪厚度，径流
- CO2、碳、氮、磷排放
- 2米气温、湿度，10米风速
- 感热、潜热、蒸发、反射率

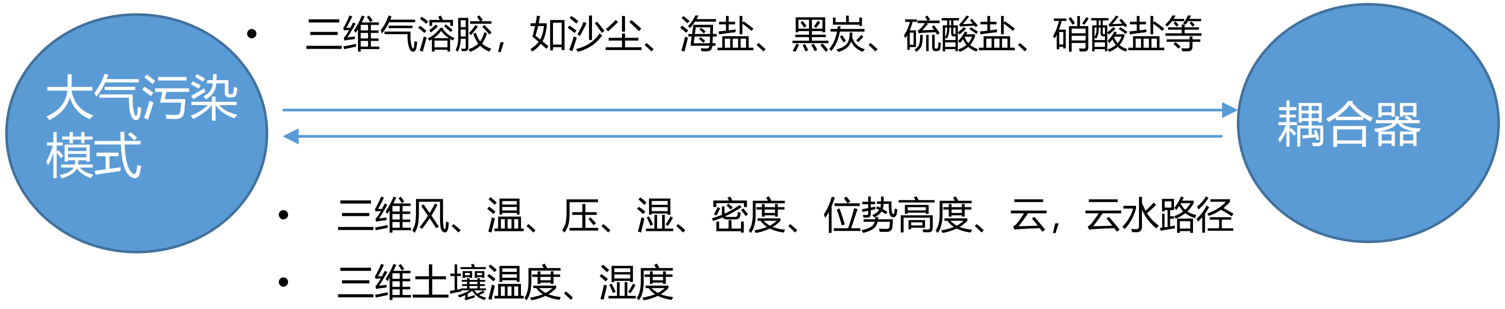


陆面模式

耦合器

- CO2，碳、氮、磷通量
- 降水、降雪
- 短波辐射、长波辐射
- 地表面气压
- 最低层风、压、温、湿度、密度

大气气溶胶收发数据



大气污染模式

耦合器

- 三维气溶胶，如沙尘、海盐、黑炭、硫酸盐、硝酸盐等

- 三维风、温、压、湿、密度、位势高度、云，云水路径
- 三维土壤温度、湿度

谢　谢